10/630174

PATENT
Attorney Docket No. 04546

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent No. 7,177,978

Confirmation No. 7653

Issued: Feb. 13. 2007

Name of Patentee: KANSAL ET AL.

Patent Title: GENERATING AND MERGING LOOKUP RESULTS TO APPLY MULTIPLE FEATURES

CERTIFICATE OF MAILING

I hereby certify that this paper is being deposited with the United States Postal Service on the date shown with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231, on August 17, 2007.

Kirk D. Williams, Esq.

REQUEST FOR CERTIFICATE OF CORRECTION OF PATENT FOR PATENT OFFICE MISTAKE (37 C.F.R. § 1.322)

Attn: Certificate of Correction Branch
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

It is requested that a Certificate of Correction be issued to correct Office mistakes found the above-identified patent. Attached hereto is a Certificate of Correction which indicates the requested correction. For your convenience, also attached are copies of selected pages (a) from the issued patent with errors highlighted, and (b) from the original application as filed July 29, 2003 and (c) Petition to Withdraw RCE and Amendment C filed July 27, 2006, with the correct text/instructions.

In re US Patent No. 7,177,978

It is believed that there is no charge for this request because applicant or applicants were not responsible for such error, as will be apparent upon a comparison of the issued patent with the application as filed or amended. However, the Assistant Commissioner is hereby authorized to charge any fee that may be required to Deposit Account No. 501430.

Respectfully submitted,
The Law Office of Kirk D. Williams

Date: August 17, 2007

By _____ Aug 17, 2007

Kirk D. Williams, Reg. No. 42,229
One of the Attorneys for Applicants
CUSTOMER NUMBER 26327
The Law Office of Kirk D. Williams
P.O. Box 61538, Denver, CO 80206-8538
303-282-0151 (telephone), 303-778-0748 (facsimile)

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,177,978

DATED : Feb. 13, 2007

INVENTOR(S) : Kanekar et al.

It is certified that error(s) appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Col. 20, line 20, replace "906I" with -- 960I --

Col. 20, line 43, replace "906I" with -- 960I --

Col. 21, line 37, replace "906I" with -- 960I --

Col. 24, line 41, replace "indentifying" with -- identifying --

MAILING ADDRESS OF SENDER:
Kirk D. Williams, Reg. No. 42,229
Customer No. 26327
The Law Office of Kirk D. Williams
P.O. Box 61538, Denver, CO 80206

PATENT NO. 7,177,978
No. of additional copies

⇨ NONE (0)

AUG 22 2007

with the entries (and data structures updated as required). Processing is complete as indicated by process block **712**.

FIG. **7B** illustrates a process for identifying a highest priority result used in one embodiment. Processing begins with process block **750**, and proceeds to process block **752**, wherein results are received from the associative memories, blocks, etc. (including possibly from previous stages). In process block **754**, the priority values are associated with the results (e.g., based on the entries, memories, blocks, etc.). In process block **756**, the highest priority result is (or in one embodiment, results are) identified based on the inherent or programmed priority values. The hierarchy (e.g., the order they are considered) of types of priority values (e.g., those associated with the entries, banks, memories, etc.) can vary among embodiments and even among individual lookup operations. In process block **758**, the highest priority result is (or results are) identified. Processing is complete as indicated by process block **759**.

FIGS. **8A–G** illustrate access control lists, processes, mechanisms, data structures, and/or other aspects of some of an unlimited number of systems employing embodiments for updating counters or other accounting devices, or for performing other functions. Shown in FIG. **8A** is an access control list **800** which defines accounting information to be collected in a counting mechanism one by statement **801** for access control list entries **803** and in a counting mechanism two by statement **802** for access control list entries **804**. Note, there are multiple access control entries in that will cause a same counting mechanism to be adjusted. Also, the value that a particular counter is adjusted can be one (e.g., corresponding to one item or packet), a byte count (e.g., a size of an item, packet, frame, or datagram) or any other value.

FIG. **8B** illustrates a process used in one embodiment to configure a mechanism for accumulating information based on access control entries. Note, this embodiment may be responsive to and/or implemented in computer-readable medium (e.g., software, firmware, etc.), custom hardware (e.g., circuits, ASICs, etc.) or via any other means or mechanism, such as, but not limited to that disclosed herein. For example, one embodiment uses a system described herein, and/or illustrated in FIGS. **1A–E**, **2**, **8D–8E**, **9A**, **9C–D**, and/or any other figure.

Processing of the flow diagram illustrated in FIG. **8B** begins with process block **810**, and proceed to process block **812**, wherein an access control list is identified. Typically, the access control list includes multiple access control list entries, with a subset of these entries identifying accounting requests. Next, in process block **814**, accounting mechanisms are associated with each of the access control list entries specifying accounting requests. Typically, but not always, at least one of the accounting mechanisms is associated with at least two different access control list entries. Processing is complete as indicated by process block **816**.

FIG. **8C** illustrates a process used in one embodiment for updating an accounting mechanism based on an item, such as, but not limited to one or more fields or values associated with a packet. Processing begins with process block **820**, and proceeds to process block **822**, wherein an item is identified. The identification of an item might include identifying an autonomous system number corresponding to the packet. Note, an autonomous system number is typically associated with a set of communication devices under a single administrative authority. For example, all packets sent from an Internet Service Provide typically are associated with a same autonomous system number. Next, in process block **824**, a particular one of the accounting mechanisms corresponding to the item is identified, such as by, but not limited to a lookup operation in a data structure, associative memory, or by any other means or mechanism. Then, in process block **826**, the identified accounting mechanism is updated. Processing is complete as indicated by process block **828**.

FIG. **8D** illustrates one embodiment of a system for updating an accounting value based on that defined by an access control list or other mechanism. Packets **831** are received and processed by packet processor **832** to generate packets **839**. In one embodiment, packet processor **832** performs a lookup operation in a forwarding information base (FIB) data structure to identify the source and/or destination autonomous system number associated with the identified packet.

Based on an identified packet, autonomous system numbers, and/or other information, a lookup value **833** is identified. FIG. **9G** illustrates a lookup value **960** used in one embodiment. One embodiment uses all, less than all, or none of fields **960A–960I**.

960I

Based on lookup value **833**, a lookup operation is performed in associative memory entries **834** in one or more associative memory banks and/or one or more associative memories to generate a counter indication **835**. The corresponding counting mechanism within counters and decoder/control logic **836** is updated. Counter values **837** are typically communicated via any communication mechanism and/or technique to packet processor **832** or another device to be forwarded or processed.

FIG. **8E** illustrates one embodiment of a system for updating an accounting value based on that defined by an access control list or other mechanism. Packets **840** are received and processed by packet processor **841** to generate packets **849**. In one embodiment, packet processor **841** performs a lookup operation in a forwarding information base (FIB) data structure to identify the source and/or destination autonomous system number associated with the identified packet.

Based on an identified packet, autonomous system numbers, and/or other information, a lookup value **842** is identified. FIG. **9G** illustrates a lookup value **960** used in one embodiment. One embodiment uses all, less than all, or none of fields **960A–960I**.

960I

Based on lookup value **842**, a lookup operation is performed in associative memory entries **843** in one or more associative memory banks and/or one or more associative memories to produce a lookup result **844**, which is then used to perform a lookup operation in adjunct memory **845** generate a counter indication **846**, and the corresponding counting mechanism within counters and decoder/control logic **847** is updated. In one embodiment, adjunct memory **845** stores counter indications for corresponding locations of access control list entries programmed in associative memory **843**, and some of these counter indications may be the same value such that a same counting mechanism is updated for different matching access control list entries. Counter values **848** are typically communicated via any communication mechanism and/or technique to packet processor **841** or another device to be forwarded or processed.

FIG. **8F** illustrates an example of associative memory entries **860** and corresponding adjunct memory entries **870**, such as those are generated by one embodiment based on access control list entries **803** and **804** (FIG. **8A**). As shown, associative memory entries **861–863** have the same counter indication in adjunct memory entries **871–873**, while associative memory entry **864** has a different corresponding counter indication in adjunct memory entry **874**. In one

embodiment, associative memory entries include fields for a source address, destination address, and other fields, such as, but not limited to autonomous system numbers (ASNs), protocol type, source and destination port information, etc. In one embodiment, adjunct memory entries **870** include an indication of a counting mechanism and/or other values which may be used for other purposes (e.g., security, routing, policing, quality of service, etc.).

FIG. **8**G illustrates a process used in one embodiment for processing a packet. Processing begins with process block **880**, and proceeds to process block **882**, wherein a packet is identified. Next, in process block **884**, one or more forwarding information base (FIB) lookup operations are performed to identify source and destination autonomous system numbers corresponding to the identified packet. In process block **886**, an accounting lookup value is identified, typically based on information contained in the identified packet and the source and destination ASNs. In process block **888**, a lookup operation is performed in one or more associative memory banks and possibly in corresponding one or more adjunct memories to identify a counter indication. In process block **890**, the counter, if any, corresponding to the counter indication is updated by some static or dynamic value. Processing is complete as indicated by process block **892**.

FIG. **9**A illustrates one embodiment of a system for identifying a merged lookup result. Packets **901** are received and processed by packet processor **902** to generate packets **909**. In one embodiment, packet processor **902** performs a lookup operation in a forwarding information base (FIB) data structure to identify the source and/or destination autonomous system number associated with the identified packet.

Based on an identified packet, autonomous system numbers, and/or other information, a lookup value **903** is identified. FIG. **9**G illustrates a lookup value **960** used in one embodiment. One embodiment uses all, less than all, or none of fields **960**A–**960**I.

960I

Based on lookup value **903**, a lookup operation is performed in associative memory entries **904** (e.g., access control list, security, quality of service, accounting entries) in multiple associative memory banks and/or one or more associative memories to generate a results **905**, based on which, memories **906** generate results **907**. Combiner mechanism **910** merges results **907** to produce one or more merged results **911**, which are typically used by packet processor **902** in the processing of packets. In one embodiment, combiner mechanism **910** includes a processing element responsive to computer-readable medium (e.g., software, firmware, etc.), custom hardware (e.g., circuits, ASICs, etc.) and/or via any other means or mechanism. In one embodiment, a merged result **911** includes a counter indication which is used by counters and decoder/control logic **912** to update a value. The accumulated accounting values **913** are typically communicated to packet processor **902** or another device.

FIG. **9**B illustrates an access control list **915**, including access control list entries of multiple features of a same type. For example, entries **916** correspond to security entries such as the packet that should be dropped or processed, while entries **917** correspond to packets that should or should not be sent to a mechanism to encrypt the packet. Different associative memories are each programmed with associative memory entries corresponding to a different one of the features. A lookup operation is then performed substantially simultaneously on each of feature sets of associative memory entries to generate associative memory results, which are then used to perform lookup operations substantially simultaneously in adjunct memories to produce the lookup results which then can be merged to produce the merged result. The respective priorities of the lookup results may be implicit based on that corresponding to their respective associative memory banks and/or adjunct memories, or be specified, such as in the associative memory entries, from another data structure lookup operation, or identified using any other manner or mechanism.

For example, one embodiment includes four associative memory banks for supporting one to four features. An associative memory lookup operation is performed in parallel on the four banks and then in the adjunct memories (SRAMs), which indicate the action, type of entry (e.g., ACL, QoS, Accounting), and precedence for combiner mechanism. The combiner mechanism merges the results to get the final merged result. A miss in an ACL lookup in a bank is treated as a permit with lowest precedence. If in more than one bank there is a hit with same specified precedence in the retrieved adjunct memory entry, the precedence used by the combiner mechanism is determined based on the implied or specified precedence of the associative memory bank. If there is a miss in all the banks, default result is used from global registers. A similar merge operation is performed for the QoS and accounting lookup results.

FIG. **9**C illustrates a lookup and merge mechanism **920** used by one embodiment. One or more of associative memory banks **921**A–**921**C (there can be any number of banks) are programmed with associative memory entries of a same access control list type, with different features of the type programmed into a different one of the associative memory banks **921**A–**921**C. Corresponding adjunct memory entries **922**A–**922**C are programmed in one or more adjunct memories. Thus, lookup operations can be performed substantially simultaneously on associative memory banks **921**A–C to generate results, which are used to identify corresponding lookup results from adjunct memory entries **922**A–**922**C, which are then merged by combiner mechanism **923** to generate the merged result **924**.

FIG. **9**D is substantially similar to that of FIG. **9**C, but illustrates that multiple merged results corresponding to multiple access control list entry types can be generated in parallel (e.g., substantially simultaneously). As shown, lookup and merge mechanism **920**, used by one embodiment, is programmed with features sets of a same type in associative memory banks **931**A–**931**B (there can be any number of banks), and of a different type in associative memory banks **931**C–**931**D (there can be any number of banks). Corresponding adjunct memory entries **932**A–**932**D are programmed into one or more adjunct memories. Thus, lookup operations can be performed substantially simultaneously on associative memory banks **921**A–D to generate results, which are used to identify corresponding lookup results from adjunct memory entries **922**A–**922**D, which are then merged by combiner mechanism **933** to generate the multiple merged results **934** (e.g., typically one or more merged result per access control list type).

FIG. **9**E illustrates a process used in one embodiment to program the associative and adjunct memories in one embodiment. Processing begins with process block **940**, and proceeds to process block **941**, wherein an access control list including multiple access control list entries is identified. In process block **942**, a first set of the access control list entries corresponding to a first feature of the access control list entries is identified. In process block **943**, a first associative memory bank and a first adjunct memory are programmed with entries corresponding to the first set of access control

list entries. In process block **944**, a second set of the access control list entries corresponding to a second feature of the access control list entries is identified. In process block **945**, a second associative memory bank and a second adjunct memory are programmed with entries corresponding to the second set of access control list entries. The first set of associative memory entries have a higher lookup precedence than the second set of associative memory entries. Processing is complete as indicated by process block **946**.

FIG. **9F** illustrates a process used by one embodiment to perform lookup operations and to identify the merged result. Processing begins with process block **950**, and proceeds to process block **951**, wherein a lookup value is identified. Next, in process block **952**, lookup operations are performed in the first and second associative memory banks and adjunct memories to generate first and second lookup results, which are merged in process block **953** to identify the merged result. Processing is complete as indicated by process block **954**.

FIG. **9G** illustrates a lookup value **960**, result value **965**, and merged result value **967** used in one embodiment. As shown, lookup value **960** includes a lookup type **960A**, source address **960B**, destination address **960C**, source port **960D**, destination port **960E**, protocol type **960F**, source ASN **960G**, destination ASN **960H**, and possibly other fields **960I**. One embodiment uses all, less than all, or none of fields **960A–960I**.

As shown, result value **965** includes a result type **965A**, an action or counter indication **965B**, and a precedence indication **965C**. In one embodiment, result value **965** is programmed in the adjunct memories. One embodiment uses all, less than all, or none of fields **965A–965C**.

As shown, merged result value **967** includes a result type **967A** and an action or counter indication **967B**. One embodiment uses all, less than all, or none of fields **967A–967B**.

FIGS. **9H–9J** illustrate merging logic truth tables **970**, **972**, and **974** for generating the merged result. In one embodiment, the merge result of a security lookup operation is illustrated in security combiner logic **970**, and is based on the results of up to four substantially simultaneous (or not) lookup operations with differing precedence indicated in columns **970A–970D**, with the corresponding merged result shown in column **970E**. Note, the "- - -" in the fields indicate a don't care condition as a merged result corresponding to a higher priority will be selected.

In one embodiment, the merge result of a Quality of Service (QoS) lookup operation is illustrated in security combiner logic **972**, and is based on the results of a previously merged security lookup operation and up to four substantially simultaneous (or not) lookup operations with differing precedence indicated in columns **972A–970E**, with the corresponding merged result shown in column **972F**.

In one embodiment, the merge result of an accounting lookup operation is illustrated in accounting combiner logic **972**, and is based on the results of a previously merged security lookup operation and up to four substantially simultaneous (or not) lookup operations with differing precedence indicated in columns **974A–974E**, with the corresponding merged result shown possibly identifying a counter to be updated in column **972F**.

FIG. **9K** illustrates a process used in one embodiment, to generate a security merged result, a QoS merged result, and an accounting merged result. Processing begins with process block **980**, and proceeds to process block **981**, wherein a packet is identified. Next, in process block **982**, one or more FIB lookup operations are performed to identify source and destination ASNs. In process block **983**, a security lookup value is identified. In process block **984**, lookup operations are performed based on the security lookup value in multiple associative memory banks and one or more adjunct memories to identify multiple security results, which are merged in process block **985** to identify the merged security result. Also, this merged security result is stored in a data structure or other mechanism for use in identifying the merged QoS and accounting results.

In process block **986**, the QoS lookup value is identified. In process block **987**, lookup operations are performed based on the QoS lookup value in multiple associative memory banks and one or more adjunct memories to identify multiple QoS results, which, in process block **988**, are merged along with the previously determined merged security result to identify the merged QoS result.

In process block **989**, the accounting lookup value is identified. In process block **990**, lookup operations are performed based on the accounting lookup value in multiple associative memory banks and one or more adjunct memories to identify multiple accounting results, which, in process block **991**, are merged along with the previously determined merged security result to identify the merged accounting result. Also, an identified counter or other accounting mechanism is updated. Processing is complete as indicated by process block **992**.

In view of the many possible embodiments to which the principles of our invention may be applied, it will be appreciated that the embodiments and aspects thereof described herein with respect to the drawings/figures are only illustrative and should not be taken as limiting the scope of the invention. For example and as would be apparent to one skilled in the art, many of the process block operations can be re-ordered to be performed before, after, or substantially concurrent with other operations. Also, many different forms of data structures could be used in various embodiments. The invention as described herein contemplates all such embodiments as may come within the scope of the following claims and equivalents thereof.

What is claimed is:

identifying

1. A method for indentifying a merged lookup result, the method comprising:

identifying an access control list including a plurality of access control list entries;

identifying a first set of access control list entries corresponding to a first feature of said plurality of access control list entries;

programming a first associative memory bank and a first adjunct memory with first associative memory entries corresponding to the first set of access control list entries

identifying a second set of access control list entries corresponding to a second feature of said plurality of access control list entries; and

programming a second associative memory bank and a second adjunct memory with second associative memory entries corresponding to the second set of access control list entries;

wherein said first associative memory entries have a higher lookup precedence than said second associative memory entries.

2. The method of claim 1, comprising:

identifying a lookup value;

performing lookup operations in the first associative memory bank and the first adjunct memory to generate a first lookup result;

From Application Filed 7-29-2003

the identified accounting mechanism is updated. Processing is complete as indicated by process block 828.

FIG. 8D illustrates one embodiment of a system for updating an accounting value based on that defined by an access control list or other mechanism. Packets 831 are received and processed by packet processor 832 to generate packets 839. In one embodiment, packet processor 832 performs a lookup operation in a forwarding information base (FIB) data structure to identify the source and/or destination autonomous system number associated with the identified packet.

Based on an identified packet, autonomous system numbers, and/or other information, a lookup value 833 is identified. FIG. 9G illustrates a lookup value 960 used in one embodiment. One embodiment uses all, less than all, or none of fields 960A-960I.

(1) error in Col. 20, line 20

Based on lookup value 833, a lookup operation is performed in associative memory entries 834 in one or more associative memory banks and/or one or more associative memories to generate a counter indication 835. The corresponding counting mechanism within counters and decoder/control logic 836 is updated. Counter values 837 are typically communicated via any communication mechanism and/or technique to packet processor 832 or another device to be forwarded or processed.

FIG. 8E illustrates one embodiment of a system for updating an accounting value based on that defined by an access control list or other mechanism. Packets 840 are received and processed by packet processor 841 to generate packets 849. In one embodiment, packet processor 841 performs a lookup operation in a forwarding information base (FIB) data structure to identify the source and/or destination autonomous system number associated with the identified packet.

Based on an identified packet, autonomous system numbers, and/or other information, a lookup value 842 is identified. FIG. 9G illustrates a lookup value 960 used in one embodiment. One embodiment uses all, less than all, or none of fields 960A-960I.

(2) Error in Col. 20, Line 43

Based on lookup value 842, a lookup operation is performed in associative memory entries 843 in one or more associative memory banks and/or one or more

From Patent Application Filed July 29, 2003

corresponding one or more adjunct memories to identify a counter indication. In process block 890, the counter, if any, corresponding to the counter indication is updated by some static or dynamic value. Processing is complete as indicated by process block 892.

FIG. 9A illustrates one embodiment of a system for identifying a merged lookup result. Packets 901 are received and processed by packet processor 902 to generate packets 909. In one embodiment, packet processor 902 performs a lookup operation in a forwarding information base (FIB) data structure to identify the source and/or destination autonomous system number associated with the identified packet.

Based on an identified packet, autonomous system numbers, and/or other information, a lookup value 903 is identified. FIG. 9G illustrates a lookup value 960 used in one embodiment. One embodiment uses all, less than all, or none of fields 960A-960I.

(3) Error in Col. 21, line 37.

Based on lookup value 903, a lookup operation is performed in associative memory entries 904 (e.g., access control list, security, quality of service, accounting entries) in multiple associative memory banks and/or one or more associative memories to generate a results 905, based on which, memories 906 generate results 907. Combiner mechanism 910 merges results 907 to produce one or more merged results 911, which are typically used by packet processor 902 in the processing of packets. In one embodiment, combiner mechanism 910 includes a processing element responsive to computer-readable medium (e.g., software, firmware, etc.), custom hardware (e.g., circuits, ASICs, etc.) and/or via any other means or mechanism. In one embodiment, a merged result 911 includes a counter indication which is used by counters and decoder/control logic 912 to update a value. The accumulated accounting values 913 are typically communicated to packet processor 902 or another device.

FIG. 9B illustrates an access control list 915, including access control list entries of multiple features of a same type. For example, entries 916 correspond to security entries such as the packet that should be dropped or processed, while entries 917 correspond to packets that should or should not be sent to a mechanism to encrypt the packet. Different associative memories are each programmed with associative memory

From Petition to Withdraw RCE and Amendment C filed 7-27-2006

Amendments to the Claims:

The listing of clams will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1-3 (cancelled)

(4) Error in Cl. 24, Line 42

Claim 4 (currently amended): A method for identifying a merged lookup result, the method comprising:

identifying an access control list including a plurality of access control list entries;

identifying a first set of access control list entries corresponding to a first feature of said plurality of access control list entries;

programming a first associative memory bank and a first adjunct memory with first associative memory entries corresponding to the first set of access control list entries

identifying a second set of access control list entries corresponding to a ~~first~~ second feature of said plurality of access control list entries; and

programming a second associative memory bank and a second adjunct memory with second associative memory entries corresponding to the second set of access control list entries;

wherein said first associative memory entries have a higher lookup precedence than said second associative memory entries.